AMENDMENTS TO THE SPECIFICATION

Please replace the title of the invention as follows:

SYSTEM AND METHOD INCLUDING DISTRIBUTED INSTRUCTION BUFFERS HOLDING A ~~SECOND~~ STATIC PREDECODED INSTRUCTION FORM

Please replace the Abstract as follows:

A system and method is provided for processing a first instruction set and a second instruction set in a single processor. The method includes storing a plurality of ~~control signals~~ instructions of the second instruction set in a plurality of buffers proximate to a plurality of execution units, ~~wherein the control signals are predecoded instructions of the second instruction set~~, executing an instruction of the first instruction set in response to a ~~branch instruction of the first instruction set~~ first counter, and executing at least one instruction ~~the control signals for an instruction~~ of the second instruction set in response to ~~a branch instruction of the second instruction set~~ at least a second counter, wherein the second counter is invoked by a branch instruction of the first instruction set.

Please replace the paragraph beginning at page 3, line 1 as follows:

According to an embodiment of the present invention, a method is provided for processing a first instruction set and a second instruction set in a single processor. The method includes storing a plurality of ~~control signals~~ instructions of the second instruction set in a plurality of buffers proximate to a plurality of execution units, ~~wherein the control signals are predecoded instructions of the second instruction set~~, executing an instruction of the first instruction set in response to a ~~branch instruction of the first instruction set~~ first counter, and

executing at least one instruction ~~the control signals for an instruction~~ of the second instruction set in response to ~~a branch instruction of the second instruction set~~ at least a second counter, wherein the second counter is invoked by a branch instruction of the first instruction set.

Please replace the paragraph beginning at page 9, line 6 as follows:

Instructions to be stored in the primary instruction memory 321 and decoded instructions (control signals) to be stored in the local predecoded instruction buffers 306-~~321~~310 can be generated by the code assignment phase of a compiler. The compiler can target the two instruction formats and issue widths. Instructions of the second format contain one bit for each of the control signals generated by the instruction decoder of the first format. Because the second format includes the predecoded form of the first format, instructions of the second format will be wider, or include more bits, than instructions of the first format. The increase in instruction width may be accompanied by an increase in execution speed as described below. The compiler places decoded blocks of machine code (e.g., a small loop which is frequently accessed) into the local predecoded instruction buffers based either on static analysis or execution profiling.

Please replace the paragraph beginning at page 10, line 8 as follows:

The local predecoded instruction buffers 306-310 are associated one-to-one, in close physical proximity, with each execution unit 301-305. Each local predecoded instruction buffer is statically loaded with decoded instructions (control signals) of the alternate instruction form. Because these local buffers are smaller than the primary instruction cache 321 and proximate to the execution unit, they can be accessed faster than the primary instruction cache 321. Proximity is a function of speed, in a processor according to the present invention, there is no significant

logic delay in fetching the decoded instructions stored in the buffers for the execution hardware. Thus, a buffer may be located at a position ~~specially~~ spatially distant from the execution hardware, however, according to the present invention, a buffer-to-execution hardware pathway with no significant logic delay as compared to the primary instruction fetch mechanism is considered proximate. An alternate fetch/issue mechanism eliminates any instruction fetch bandwidth limitation.

Please replace the paragraph beginning at page 12, line 3 as follows:

After the sequencer 325 is invoked, it switches a plurality of gates 316-320 prior to the execution queues 301-305 for the primary instruction form, de-gating the primary instruction form. In addition, the branch unit 305 signals the fetch unit 322 to stop fetching instructions of the primary form from memory 321. The sequencer 325 includes an alternate program counter for directing the fetching of the decoded instructions. Further, the sequencer 325 sequences ~~of~~ the decoded instructions (control signals) from the local predecoded instruction buffers. Individual program counters can be implemented for each buffer to improve the efficiency with which the buffer space is used.